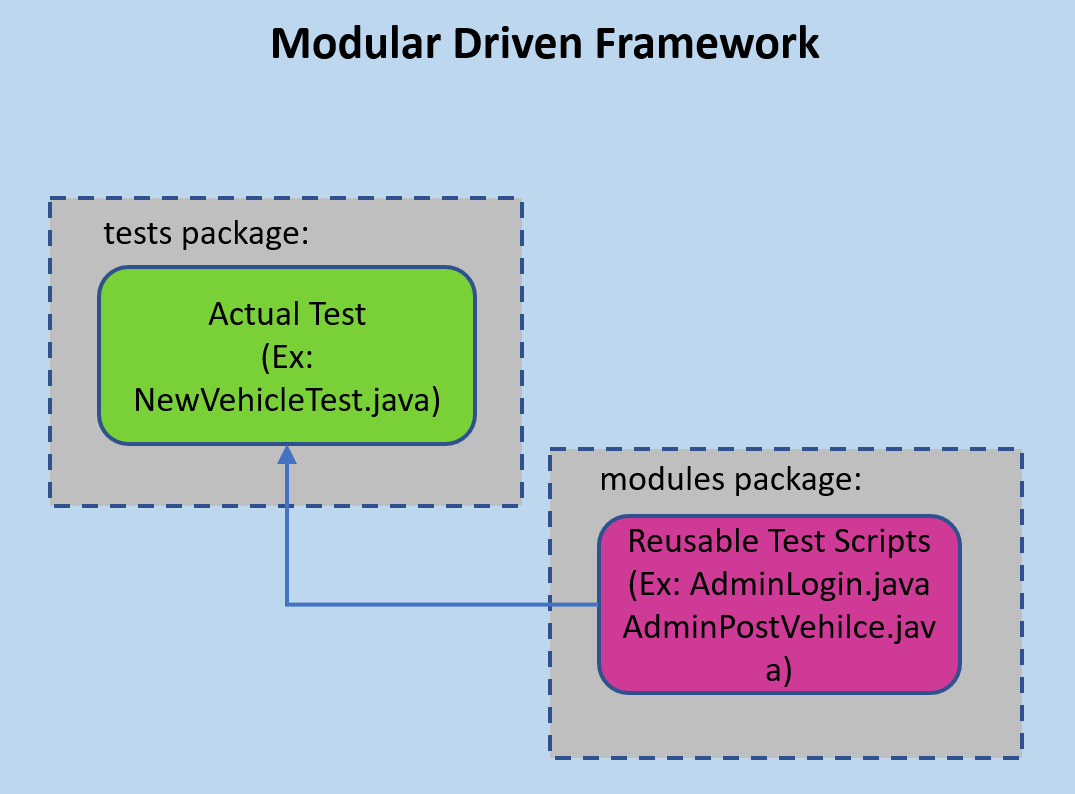
**Selenium Modular Framework:**

In most of the applications there are functionalities which are repeated in most of the tests. For example, a login functionality might be required for all the tests.

In a modular driven framework, we write separate test scripts for these common functionalities and use them in other tests where ever needed.



In the sample application that we are going discuss, which is an online car rental system, the following are some of the functionalities that are frequently used.

1. Admin Login
2. Admin posting a vehicle
3. Admin posting a brand etc.

When we automate the system using Modular Driven Framework, we separate this functionality in the form of separate test scripts and use them in other test scripts.

Pros:

1. The framework introduces the high level of modularization which leads to easier and cost-efficient maintenance.
2. The framework is pretty much scalable
3. If the changes are implemented in one part of the application, only the test script representing that part of the application needs to be fixed to leave all the other parts untouched.

Cons:

1. While implementing test scripts for each module separately, we embed the test data (Data with which we are supposed to perform testing) into the test scripts. Thus, whenever we are supposed to test with a different set of test data, it requires the manipulations to be made in the test scripts.